Terms used **java serialization interface** Found **530** of **780,315**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The Digital Library

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 Object serialization for marshalling data in a Java interface to MPI
Bryan Carpenter, Geoffrey Fox, Sung Hoon Ko, Sang Lim
June 1999 **Proceedings of the ACM 1999 conference on Java Grande**
Full text available: pdf(688.78 KB) Additional Information: full citation, references, citings, index terms

2 Java:introduction
Vishal Shah
November 1997 **Crossroads**, Volume 4 Issue 2
Full text available: html(37.68 KB) Additional Information: full citation, index terms

3 Java virtual machine support for object serialization
Fabian Breg, Constantine D. Polychronopoulos
June 2001 **Proceedings of the 2001 joint ACM-ISCOPE conference on Java Grande**
Full text available: pdf(617.30 KB) Additional Information: full citation, abstract, references, citings

Distributed computing has become increasingly popular in the high performance community. Java's Remote Method Invocation (RMI) provides a simple, yet powerful method for implementing parallel algorithms. The performance of RMI has been less than adequate, however, and object serialization is often identified as a major performance inhibitor. We believe that object serialization is best performed in the Java Virtual Machine (JVM), where information regarding object layout and hardware communic ...

4 Ibis: an efficient Java-based grid programming environment
Rob V. van Nieuwpoort, Jason Maassen, Rutger Hofman, Thilo Kielmann, Henri E. Bal
November 2002 **Proceedings of the 2002 joint ACM-ISCOPE conference on Java Grande**
Full text available: pdf(120.05 KB) Additional Information: full citation, abstract, references, index terms, review

In computational grids, performance-hungry applications need to simultaneously tap the computational power of multiple, dynamically available sites. The crux of designing grid programming environments stems exactly from the dynamic availability of compute cycles: grid programming environments (a) need to be *portable* to run on as many sites as possible, (b) they need to be *flexible* to cope with different network protocols and dynamically changing groups of compute nodes, while (c) t ...

Keywords: Java, grid computing, performance, portability